SPECIFICATION

TITLE OF THE INVENTION

METHOD AND APPARATUS FOR MOTION COMPENSATION ADAPTIVE IMAGE PROCESSING

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a compression of decoded picture and particularly relates to a method and an apparatus for motion compensation adaptive image processing of the decoded picture data,which is not limited to but can be used in video decoding systems such as ones adopting MPEG video coding standards.

2. Description of the conventional Art

In order for a real-time VLSI decoder to be cost effective, it is necessary to reduce its resources. There are several ways to reduce the cost but one of them is to use less memory. Other methods could reduce computational complexity, lower its bandwidth usage, and memory more which include implementation issues.

The Advanced Television System Committee (ATSC) adopted the ISO/IEC 18318-2 a.k.a. the MPEG-2 Video Coding Standard which encodes and decodes moving pictures. Specially, the Digital Television (DTV) System in the U.S. adopts MPEG-2 MP@HL video coding specification as its standard and specifies 18 different picture formats. Among the many formats, 1920 $\times$ 1082 $\times$ 30 frames/sec requires the most frame memory and bandwidth. To decode this picture size, a decoder requires about 16 Mbytes of memory since external memory is available in specific sizes.

In designing a real-time VLSI decoder, the memory bandwidth is very critical and increases the cost of the chip. Also, due to its large external memory requirement for decoded frame storage it will also increase the cost of its target application, such as PC add-in card or Set-top Box (STB). Therefore, above mentioned factors become critical, making a VLSI implementation inappropriate for consumer electronics.

An HDTV decoder must be able to decode all the 18 formats recommended by the ATSC specification, ATSC DTV video formats, using MPEG Main Profile/Main Level specification, are all in 4:2:0 Y, Cb, Cr mode. The 18 formats are combinations of the followings:

1) 4 different picture sizes, which are, $1920 \times 1080$, $1280 \times 720$, $704 \times 480$ and $640 \times 480$

2) 2 different aspect ratio information, which are 4:3 and 16:9

3) 8 different frame rate codes, which are 23.976 Hz, 24 Hz, 29.97Hz, 30Hz, 59.94 Hz, and 60Hz

4) progressive or interlaced sequences

Also, the MPEG-2 video adopted by the ATSC supports 3 different coding modes, which is also called picture coding types. Each of these 3 picture coding types has different characteristics. They are Intra (I), Predictive (P) and Bi-directional (B) picture types. I pictures are coded without reference to other pictures. And they can be used to predict the P and B pictures. P pictures are coded using previous I picture or other P pictures as a reference, and the P pictures can predict the future P and B pictures. B pictures are coded using I and P pictures from previous and future pictures. But B pictures are not used as a reference.

Figure 1 is a schematic block diagram of an image processing apparatus in accordance with the conventional art.

The image processing apparatus in accordance with the conventional art includes a variable length decoder 101 decoding input image data (input bitstreams) as variable length and

outputting decoded image data and motion vector signal, a motion compensation unit 106 receiving the motion vector signal from the variable length decoder 101 and outputting motion compensation information to compensate the received image data, a dequantizer 102 dequantizing the variable length decoded image data and generating dequantized image data, inverse discrete cosine transform (IDCT) unit 103 transforming the dequantized image data into inverse discrete cosine transform data, an image frame processing unit 104 processing the inverse discrete cosine transform data as a frame unit according to the motion compensation information, and a memory 105 storing the image data received from the image frame processing unit 104 and outputting the stored image data to the motion compensation unit 106 and a display (not shown).

The decoding procedure of the MPEG-2 Video is in the following order as shown in Figure 1. Input bit stream is first Variable Length Decoded at the Variable Length Decoder (VLD) 101. From the Variable Length Decoder 101 motion vector information and variable length decoded (VLD) data are outputted to the dequantizer 102. That is, the VLD data is inverse-scanned and dequantized then fed to the IDCT block. Motion vector signals are used to retrieve block data from the reference picture by the Motion Compensator. Finally, the image frame processing unit 104 processes the IDCT data based on the motion compensation information from the Motion Compensator 106 and outputs motion compensated data to reconstruct the input image data. Then, 16 MB of external memory 105 stores 3 different pictures, 2 pictures being reference and the other being the B pictures. The Display Controller (not shown) reads pictures from the external memory 105 and displays them on the TV or monitors (not shown).

The Display Controller and Motion Compensator 106 are implemented as theblocks that use most of the bandwidth. Especially for Motion Compensator 106, unlike Display Controller, the memory data needs to be accessed randomly for motion compensation purpose since, motion

compensation is done on block i.e., 16 × 16 or 16 × 18 pel basis.

Usually, MPEG encodes sequences using combination of I, P and B pictures causing the prediction error to propagate until the error is refreshed by the next I picture. Therefore, the compression scheme must be balanced between random accessibility of the decoded data and moderate compression ratio to meet the compression needs, and also be able to propagate less error.

## SUMMARY OF THE INVENTION

The object of the present invention is to provide a method and an apparatus for motion compensation adaptive image processing which is capable of compressing decoded image data as coding them on the low bandwidth, storing them in small memory and restoring the input image data to display high definition quality.

The another object of the present invention is to provide a method and an apparatus for motion compensation adaptive image processing, which is capable of analyzing input image data into image data having relatively high and low frequency components, allocating bits to the analyzed image data, compressing the image data as small and restoring the compressed image data to maintain high definition quality.

To achieve the objects, the present invention provides a method for motion compensation adaptive image processing, which processes an image data received from a external source, stores the processed image data and restores the stored data to the received image data, wherein the method comprising the steps of analyzing the received image data into image data having relatively high frequency component and relatively low frequency component, compressing/coding the image data having relatively high frequency component and relatively low frequency component which predetermined bits are allocated thereto, dividing the compressed/coded image data into a value corresponding to relatively high frequency component

and a value corresponding to relatively low frequency component, and decoding the values and restoring the received image data based on the decoding image data.

To achieve the above objects, the present invention provides an apparatus for motion compensation adaptive image process, which processes an image data received from a external source, stores the processed image data and restores the stored data to the received image data, wherein the apparatus comprises an image frame processing unit for processing the received image data as a frame unit and outputting the processed image data and a motion vector signal, an image compensating unit for generating a motion compensation information to compensate the received image data based the motion vector signal and outputting it to the image frame processing unit, an image compressing unit for analyzing image data having relatively high frequency component and image data having relatively low frequency components, allocating a predetermined bits into the analyzed image data and compressing/coding the image data including the allocated bits, a storing unit for the compressed/coded image data, and an image restoring unit for decoding the stored image data and restoring the received image data based on the decoding image data.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic block diagram of an image processing apparatus in accordance with the conventional art.

Figure 2 is a schematic block diagram of an apparatus for motion compensation adaptive image process in accordance with the present invention.

Figure 3A is a schematic block diagram to explain a concept of image compression method in accordance with the present invention.

Figure 3B is depicted in detail a schematic block diagram of compressor of an apparatus for motion compensation adaptive image process in accordance with the present invention.

Figure 4A is a schematic block diagram to explain a concept of image decompression method in accordance with the present invention.

Figure 4B is depicted in detail a schematic block diagram of decompressor of an apparatus for motion compensation adaptive image process in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Video frames contain lots of correlated regions which can be compressed. Since the frames are progressive or interlaced, compression done on horizontal direction produces far less artifacts after decompression compared to applying compression on vertical direction or 2-Dimentional direction. The higher the compression ration goes up, the more the high frequency components are hurt and edges become corrupted. DPCM, Adaptive DPCM (ADPCM) and Vector Quantization methods could corrupt the edges even though they perform well in moderate compression ratios.

Therefore, it becomes clear that an edge or high frequency preserving compression methods are necessary in order to keep the quality of the decompressed video frames. Also in texture regions where the details of the images are inherent, the previous methods tend to smooth out the detail and cause error propagation or prediction drifts.

There were also other methods which reduced these artifacts but were much more complicated since they used adaptive quantization and variable length encoding for recompression. In ADPCM method, a separated header information was stored for random access purposed for motion compensation, increasing the complexity of the decoder. The Wavelet Transform applied here keeps the complexity lower than the ADPCM method while producing near lossless visual quality, even on the texture or edge regions.

The basic concept of the present invention is about recompressing the decoded frames

before storing them to the external frame memory. It reduces the amount of storages necessary by compressing the decoded I, P, and B pictures. It also reduces the bandwidth necessary in reading and writing the frame data stored in the memory for motion compensation and display purposes. The bandwidth goes up when decoding the High Definition (HD) video sequences. Especially, when the motion compensation block is activated. Motion compensation block has to access data by $16\times8$ or $16\times16$ pel basis at a random position given by the motion vector obtained from the bitstreams. Thus these effects must be given a full consideration when designing a real-time VLSI.

Since, the MPEG uses I, P, and B pictures for encoding motion pictures, the motion estimation error propagates from an I picture to next I picture when recompression is adopted. Therefore, it is necessary to build a compression system which minimizes error propagation. Reducing memory access bandwidth along with memory size is very critical in consumer electronics since they are directly related to product cost. If we reduce the bandwidth then we can use cheaper and common synchronous DRAM (SDRAM) for the product instead of using comparatively expensive Rambus DRAM (RDRAM). The present invention keeps a good balance between the compression ratio, bandwidth usages, random accessibility for motion compensation, error propagation and cost effectiveness of the product at no visual degradation or negligible degradation in PSNR wise.

[Embodiment]

The wavelet transform used in the compression has excellent energy compacting capability. It decomposes the high frequency and low frequency components into 2 different frequency subbands, where entropy of the decomposed high frequency coefficients is reduced therefore enabling a CODEC to use less bits for encoding. Also, more high frequency components become zero or near zero value by quantization thus introducing only few number

of non zero values in the high frequency subband. This high frequency subband can be efficiently coded via simple quantization, quantization using preconstructed table, vector quantization or variable length coding. Compression ratio can be controlled using quantization step size and it can also be frequency subbands forming low-low, low-high, and high subbands. This kind of continuous decomposintion of the low frequency component brings down example of a continuous decomposition.

Figure 2 is a schematic block diagram of an apparatus for motion compensation adaptive image processing in accordance with the present invention.

The apparatus for motion compensation adaptive image processing in accordance with the present invention, which processes an image data received from a external source, stores the processed image data and restores the stored data to the received image data, comprises a variable length decoder 200 for decoding the image data (bitstreams) to have variable length and outputting variable length decoded image data and motion vector signal, an image compensating unit 800 for generating a motion compensation information to compensate the received image data based the motion vector signal, a dequantizer 300 for dequantizing the variable length decoded image data, an inverse discrete cosine transform (IDCT) unit 400 for inverse discrete cosine transforming the variable length decoded image data , an image frame processing unit 500 for processing the inverse discrete cosine transformed data based on the motion compensation information and outputting the processed image data as a frame unit, an image compressor 600 for receiving the processed image data from the image frame processing unit 500, analyzing the image data having relatively high frequency component and image data having relatively low frequency components, allocating a predetermined bits into the analyzed image data and compressing/coding the image data including the allocated bits, a memory 700 storing the compressed/coded image data, and an image decompressor 900 decoding the stored image data, restoring the received image data based on the decoded image data and outputting the restored

image data to the motion compensation unit 800 and an external display (not shown).

Figure 3A is a schematic block diagram to explain a concept of image compression method in accordance with the present invention.

The image compressor 600 comprises high pass filters 601, 605 and 609for high pass filtering the image data received from the image frame processing unit 500 and analyzing image data having high frequency components, low pass filters 602, 606 and 610for low pass filtering the image data received from the image frame processing unit 500 and analyzing image data having low frequency components, and decimeters 603, 604, 607, 608, 611, and 612for down sampling the outputs of the filters.

Here, input image data X(n) is divided into high frequency component and low frequency component by a high pass filter 601 and a low pass filter 602 . And decimeters 603 and 604 down-sample each output of the filters 601 and 602. Assuming the above process as one stage, i.e. stage 1, the image data having low frequency component is further divided into high and low frequency components by a high pass filter 605 and a low pass filter 606 and down-sampled by decimators 607 and 608. Therefore, the input image data X(n) is processed by the several stages as mentioned above, for example, stage 1, stage 2, and stage 3,... as shown in Figure 3A.

As shown, in each stage the processed image data is continuously processed by the low pass filter and the decimeter.

Accordingly, the apparatus for motion compensation adaptive image process in accordance with compresses image data, especially to the low bandwidth.

Figure 3B is depicted in detail a schematic block diagram of compressor of an apparatus for motion compensation adaptive image process in accordance with the present invention.

Sub a' The filter (613) of the image compressor analyzes the processing image data into image data ~~having high frequency~~ components, Hn, and low frequency components, L0 and L2. Then

the image data having high frequency components is coded based on a coding table 614, the image data having first low frequency components L0 is transmitted without processing as a code, and, regarding the image having second low frequency components L2, a code coded the result of subtracting the image data having second low frequency components L2 from the image data having the second frequency components is outputted from the table 616. That is, when the image data is mapped its value into a range of the coding tables, 614 and 615, a index corresponding to the range is outputted from the coding tables, 614 and 615, as a code.

Accordingly, the input image data are analyzed high frequency components and low frequency components and effectively compressed and coded.

It will now be explained the restoration of the image data with Figure 4A and 4B.

Figure 4A is a schematic block diagram to explain a concept of image decompression method in accordance with the present invention.

The image decompressor 900 comprises inverse decimeters 901, 902, 905, 906, 909, and 912for upsampling the compressed/coded image data stored in the memory 700 and high pass and low pass filtering the upsampled image data by the high pass filters 903, 907 and 911 and the low pass filters 904, 908, and 912, respectively.

Here, output image data $\hat{X}(n)$ generated after processing the compressed/coded image data stored in the memory 700 are upsampled by inverse decimeters 901 and 902. The upsampled image data are divided into high frequency component and low frequency component by a high pass filter 903 and a low pass filter 904 . Assuming the above process as one stage, i.e. stage 1, the image data having low frequency component is further upsampled by inverse decimators 905 and 906 and divided into high and low frequency components by a high pass filter 907 and a low pass filter 908. Therefore, the output image data $\hat{X}(n)$ is processed by the several stages as mentioned above, for example, stage 1, stage 2, and stage 3,... as shown in Figure 4A.

Figure 4B is depicted in detail a schematic block diagram of decompressor of an apparatus for motion compensation adaptive image process in accordance with the present invention.

The image data having high frequency components of the compressing/coding image data stored in the memory 700 is restored referring to a coding table 913 of the decompressor 900 and the image data having low frequency components is restored referring to a coding table 914. That is, the coding table 913 is to restore the image data having high frequency components by decoding and outputting a representative value Hn' corresponding to the index of the image data $H_{index}$ which is outputted from the coding table 614, as a code, shown as Figure 3B.

Also, the image data having first low frequency components L0 is transmitted without processing to the external display and, regarding to the image data having second low frequency components L2, when it is coded at the coding table 914 and outputted a representative value, the adder 915 adds the representative and the image data having first low frequency components and outputs the added result L2'.

Accordingly, the decompressor of the apparatus in accordance with the present invention restore the image data.

The operation of the present invention is explained in detail as follows:

The reconstruction of the signal is an exact inverse procedure of the decomposition and is shown in Figures 4A and 4B, where the $L, H, \tilde{L}$ and $\tilde{H}$ are decomposition and reconstruction filter set. For, 2 tap filters, the coefficients are given as L ={$C_o$, $C_1$, $C_2$, -$C_3$} and $H, \tilde{L}$ and $\tilde{H}$ are achieved as follows, where n ={0, 1, 2, 3}.

$$H(n) = (-1)^n L(3-n) \quad (1)$$

$$\tilde{L}(n) = L(3-n) \quad (2)$$

$$\tilde{H}(n) = H(3-n) \quad (3)$$

The following are the coefficients for the Daubechie's 4 tap compact filters for decomposition and reconstruction.

$$L(n) = \{C_o, C_1, C_2, -C_3\} \quad (4)$$

$$H(n) = \{-C_3, -C_2, C_1, -C_0\} \quad (5)$$

$$\widetilde{L}(n) = \{-C_3, C_2, C_1, C_0\} \quad (6)$$

$$\widetilde{H}(n) = \{-C0, C1, -C2, -C3\} \quad (7)$$

Theoretically, image is sub-divided based on a block size of $m \times n$, where m is the number of pixels in the horizontal direction and n is the number of lines in the vertical direction. Compression can be performed on the pre-divided sub-block basis for luminance, Cb and Cr chrominance components for I, P and B pictures.

Let X be a vector to be coded in a digital picture where the picture size is $H \times V$, where H is the horizontal size and V is the vertical size of the picture. The size of the vector can be $m \times n$, where m is the horizontal size and the n is the vertical size of the vector. The compression will be performed first on the luminance block and then on the chrominance block.

As stated before, compression ratio can be varied by controlling the decomposition layer shown in Figures 3A and 3B. Compression ratio can also be modified by applying different quantization factor for the low and high frequency coefficients. High compression ratio of low frequency components is less desirable since they tend to cause more error as the signal is reconstructed.

The quantizer can be designed in many different ways. One method could be to analyze the statistics of the sub-blocks after decomposition. The statistics would look something like the Laplacian. And optimal non-linear quantizer based on the statistics can be designed. A non-linear

quantizer is very useful when used with the transform type of compression method. Using the wavelet transform, we are able to compact more energy to low frequency coefficients and less energy exits on the high frequency coefficients enabling us to use also liner quantizer. Therefore, both linear and non-linear quantizer can be used, since most of the energy is already compactly incorporated in the low frequency coefficients.

To facilitate easy pixel access for random access application such as motion compensation error and error drift, an example using wavelet transform with compression ratio of 25%, i.e., 4:3 compression is given as an example.

Figure 2 shows where this compression algorithm can be applied in a real-time VLSI MPEG decoder.

The first step of the compression is to apply Wavelet Transform on the block of 4pixels, i.e. the sub-block size of the compression set is $4 \times 1$ pels. The compression is done horizontally since the input source can be both progressive and interlaced, so compressing horizontally could cause less error to occur. Also, vertical compression is not preferred for display purposes since display would read the data horizontally and top to bottom from the stored frame memory. The ratio of 4:3 given here is an example can be achieved easily by coding 32bits of data with 23 bits. The second step is to apply Wavelet Transform on 4pixels. The wavelet transform is a convolution process of the pixel data and wavelet filters given above. L and H are analysis filters an $\widetilde{L}$ and $\widetilde{H}$ are reconstruction filters. Wavelet decomposition is done as follows and is also depicted in Figures 3A and 3B.

$$X_{low}(n) = \sum_{k} f(n-2k)L(k) \qquad (8)$$

$$X_{high}(h) = \sum_{k} f(n-2k)H(k) \qquad (9)$$

where, n = 0, 1, 2, 3 and f() is the block of data to be decomposed.

Multiresolution decomposition using Wavelet Transform and decimation by a factor of 2 would give 2 coefficients of low frequency components and 2 coefficients of high frequency components.

The third step would be to quantize the 2 low and 2 high frequency components are average of 7 bits for low frequency components and 5 bits each for high frequency components. For the first low frequency component is encoded with 8 bits and second low frequency component is encoded in 6 bits. In the example given here, the low frequency components and high frequency components are all double precision data. Since, the first low frequency components must be between 0 and 255 to be saved in 8 bit data, it is rounded to the nearest integer as follows.

$$Y_{low}(n) = (\text{int})\{X_{low}(n) + a\} \qquad (10)$$

where, $X_{low}(n)$ is the decomposition result and $Y_{low}(n)$ is the quantized value. A is –0.5 when $Y_{low}(n)<0$ and 0.5 when $Y_{low}(n)>0$. After the rounding operation, saturation is performed on $Y_{low}(n)$, i.e.

$$Z_{low}(n) = \begin{cases} 255 & \text{if } Y_{low}(n) > 255 \\ 0 & \text{if } Y_{low}(n) < 0 \\ Y_{low}(n) & \text{otherwise} \end{cases} \qquad (11)$$

The low frequency components use DPCM method to allocate 8 and 6 bit each for the first and second low frequency component. Prediction is a table lookup process for the second low frequency component.

$$Z'_{low}(2) = Z_{low}(2) - Z_{low}(0) \tag{12}$$

and $Z'_{low}(n)$ is assigned with a predefined code using 6 bits.

The high frequency components are now quantized. The quantization performed her can be considered as a combination of non-linear quantization and linear quantization. Since, the high frequency components must fall between 127 and –128 simple rounding operation is performed first on the high frequency components.

$$Y_{high}(n) = (\text{int})(X_{high}(n) + a) \tag{13}$$

where, $X_{high}(n)$ is the decomposition result and $Y_{high}(n)$ is the quantized value. A is –0.5 when $Y_{high}(n)<0$ and 0.5 when $Y_{high}(n)>0$. After the rounding operation, saturation is performed on $Y_{high}(n)$, i.e.

$$Z_{high}(n) = \begin{cases} 127 & \text{if } \mathrm{Y_{high}}(\mathrm{n}) > 127 \\ 0 & \text{if } \mathrm{Y_{high}}(\mathrm{n}) < -128 \\ Y_{high}(n) & \text{otherwise} \end{cases} \tag{14}$$

Then, each $Z_{high}(n)$ is quantized to 5 bits using a different pre-defined quantization table.

The reconstruction is very simple, and it is an inverse procedure of the decomposition process. The reconstruction procedure is depicted in Figure 3 and is as follows. The first step is to dequantize $Z_{high}(n)$ values by a table lookup. An 8bit data is recovered for each high frequency components. The first low frequency coefficients are already in 8 bits, and second low frequency component is reconstructed by a table lookup process, which is $\widetilde{Z}'_{low}(2)$, and adding $\widetilde{Z}'_{low}(2)$ to reconstructed value to the first low frequency coefficient, $\widetilde{Z}_{low}(0) = Z_{low}(0)$, and performing wavelet transform.

$$\widetilde{Z}_{low}(2) = \widetilde{Z}'_{low}(2) + \widetilde{Z}'_{low}(0) \tag{15}$$

Inverse wavelet transform is performed on $\widetilde{Z}'_{low}(n)$ and $\widetilde{Z}'_{high}(n)$ by using $\widetilde{L}$ and $\widetilde{H}$ filters respectively. Interpolated, 'zero' values before convolution. The low frequency components are reconstructed from convolution of $Z_{low}(n)$ and $\widetilde{L}$, and high frequency components are reconstructed from convolution of $Z_{high}(n)$ and $\widetilde{H}$.

$$\widetilde{Y}_{low}(n) = \widetilde{Z}_{low}(n-k)\widetilde{L}(n-k) \tag{16}$$

$$\widetilde{Y}_{high}(n) = \widetilde{Z}_{high}(n-k)\widetilde{H}(n-k) \tag{17}$$

After the $\widetilde{Y}_{low}$ and $\widetilde{Y}_{high}$ are reconstructed, these two values are summed and multiplied by factor of 2. This multiplication factor of 2 is due to upsampling factor which is 2.

$$\widetilde{X}(n) = 2(\widetilde{Y}_{low}(n) + \widetilde{Y}_{high}(n)) \tag{18}$$

Finally, after the $\widetilde{X}(n)$ is reconstructed from $\widetilde{Y}_{low}$ and $\widetilde{Y}_{high}$, it is rounded to the nearest integer value and saturated to lie between 0 and 255.

$$\widetilde{X}(n) = (ing)\begin{cases} \widetilde{X}(n) + 0.5 & \text{if } \widetilde{X}(n) > 0 \\ \widetilde{X}(n) - 0.5 & \text{if } \widetilde{X}(n) < 0 \\ \widetilde{X}(n) & \text{otherwise} \end{cases} \tag{19}$$

$$\widetilde{X}(n) = \begin{cases} 255 & \text{if } \widetilde{X}(n) > 255 \\ 0 & \text{if } \widetilde{X}(n) < 0 \\ \widetilde{X}(n) & \text{otherwise} \end{cases} \tag{20}$$

Therefore, in this patent, a video frame compression method to differentiate low and high frequency components to optimal compress pixel data to facilitate random access, minimize error propagation, with a moderate compression ratio for a real-time VLSI implementation has been shown.